Sub Topic: Reaction Kinetics

SAND2021-4169C

12th U. S. National Combustion Meeting
Organized by the Central States Section of the Combustion Institute
May 24–26, 2021 (Virtual)
College Station, Texas

TChem - An Open Source Computational Chemistry Software Library for Heterogenous Computing Platforms

O. Diaz-Ibarra[1,*], *K. Kim*[2], *H.N. Najm*[1], *and C. Safta*[1]

[1] *Combustion Research Facility, Sandia National Laboratories, Livermore, CA*
[2] *Center for Computing Research, Sandia National Laboratories, Albuquerque, NM*
[*] *Corresponding author:* `odiazib@sandia.gov`

Abstract: TChem is a software library for solving complex chemistry problems on modern heterogeneous computing platforms. The software provides support for gas-phase and surface chemistry, thermodynamic properties, reactor models such as homogeneous gas-phase ignition, continuously stirred tank reactor, and plug-flow reactor. To exploit modern computing platforms, TChem considers batch parallelism for computing thermodynamic properties and chemical source terms simultaneously when the evaluations are required for a large scale sensitivity analysis or spatial grid points of a reactive flow problem. In this work, we present the library infrastructure and demonstrate the parallel performance of the code on computing platforms consisting of multi(many)-core architectures with GPU accelerators.

Keywords: ***TChem, Kokkos, GPU, Sensitivity Analysis, Flow Chemistry,***

1. Introduction

To perform accurate combustion analysis, it is often required to solve detailed chemical kinetics, which includes a list of elementary reactions and the reaction rate parameters. However, solving the kinetic problem is very compute intensive. For example, GRI Mech i.e., detailed kinetic model for methane combustion problems requires to process 325 reactions. Hence, including the kinetic model in the combustion analysis increases the computational cost significantly as each spatial grid point requires to solve the complex chemical kinetic model. While modern parallel computing platforms offer several orders of magnitude higher computing power, the existing software tools [1, 2] for solving the kinetic problems are not ready for the heterogeneous computing platforms such as GPUs. In this work, we present TChem [3], a performance portable parallel software toolkit for solving computational chemistry kinetic problems.

TChem is adequate for applications that involve evaluation of thermodynamic properties, chemical source terms, and Jacobians matrices for several canonical source term configurations. The code primarily considers batch parallelism solving many instances of the chemical kinetic problems, which arises from a large scale sensitivity analysis or reactive flow problems. To achieve performance portability on multiple computing platforms, we use `Kokkos` [4] as our parallel programming model. Kokkos provides high level device abstractions with multiple device specific back-end implementations such as OpenMP for CPUs, AMD HIP and NVIDIA CUDA. This allows for us to write a single version of the code and enables the code to run on multiple heterogeneous computing devices.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

To demonstrate TChem, we will consider two use cases. The first one is related to computing chemical sources terms and thermodynamic properties pertaining to reacting flow applications where multiple evaluations are required on different spatial points per iteration. The other use case is to solve multiple initial values problems in parallel. This scenario is relevant for sensitivity analysis and uncertainty quantification analyses where multiple input samples needs to be solved. To demonstrate performance portability, we will compare the performance of our implementation on Intel Xeon Gold 62348R 2x28 @2.2GHz and the NVIDIA GPU V100.

2. Overview of TChem

The TChem software is a toolkit for computing thermodynamic properties, source term and Jacobian for chemical kinetic models that involve gas and surface reactions. TChem uses NASA polynomials [5] to compute thermodynamic properties such as heat capacity and enthalpy as polynomial functions of the thermodynamic temperature. It also provides functions to evaluate the internal energy, entropy, and Gibbs free energy.

The net production rates for gas-phase or surface species are evaluated as a sparse sums over the reaction rates of individual reactions. TChem supports several formulation for reaction rate expression including enhanced third-body efficiencies and formulations for pressure-dependent reactions. Currently, TChem supports elementary, stick coefficients, and coverage dependency for surface chemistry models. The source terms for the canonical reactors provided as examples in TChem are constructed using interfaces for thermodynamic properties and individual species production rates.

Parallelism TChem is designed to exploit hierarchical parallelism to effectively map independent work items to massively concurrent compute units. The hierarchical execution model of TChem on CUDA is illustrated in Figure 1. One can first consider solving multiple kinetic models for sensitivity analysis w.r.t. different model parameters e.g., the specification for the rate constants or the auxiliary reaction parameters. The independent model problems are considered coarse grained and they are mapped to multiple Kokkos execution space instances. On CUDA, this corresponds to CUDA streams. Solving each kinetic model, multiple initial value problems are considered with varying input conditions e.g., temperature, pressure and mixture configurations. A device parallel kernel is launched using `Kokkos::parallel_for`. A group of team threads is assigned to solve a single instance of the problems. On CUDA, a grid of thread blocks are assigned to different initial value problems. Finally, the multiple concurrent threads in a team (thread block) process independent reactions and species using nested parallel constructs of `Kokkos::TeamThreadRange` and `Kokkos::ThreadVectorRange`.

Jacobian Matrices TChem evaluates the analytical and numerical derivatives for the chemical source terms functions w.r.t. the state space using TINES library [6]. TINES evaluates the analytic Jacobians using automatic differentiation (AD) tool provided by the SACADO library [7]. SACADO is template-based `C++` library overriding math operations with their known derivatives and the derivatives are updated via chain rules. The SACADO library uses Kokkos and it is portable to GPUs. TChem provides the source term functions that allows the SACADO AD types as its value type. Evaluating the function with AD types, the derivative of the function are automatically

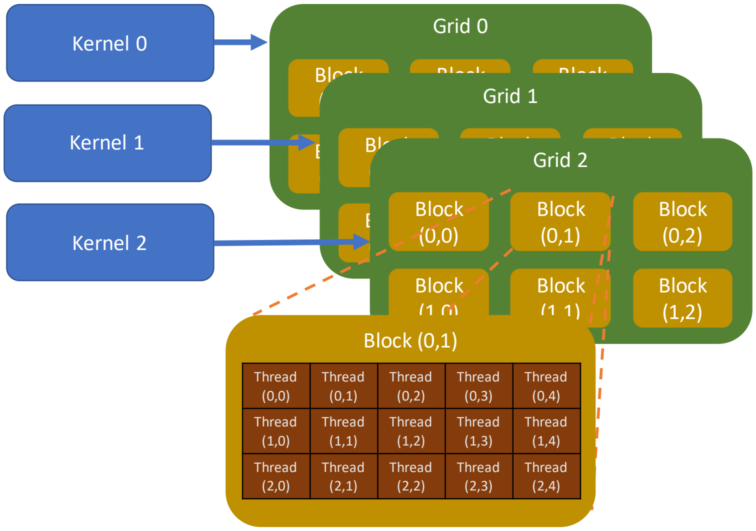

Figure 1: CUDA hierarchically parallel execution scheme. Multiple kernels can be launched for model parallelism running the simulation independently. Each device kernel is mapped to a grid of thread blocks where each thread bock is dedicated for solving a single problem in the batch parallel execution. Concurrent threads in a block is used to update the state vector and multiple independent reactions in parallel.

computed together. While numerical Jacobians require the evaluation of source term functions multiple times perturbing the input state vector, the analytic Jacobian requires only one function evaluation with a SACADO input state vector. The analytic Jacobians using SACADO greatly improves maintainability and productivity of the software as it does not require time consuming and error-prone hand derivation. TINES also provide numerical derivatives of the chemical source terms using the adaptive scheme described by Salane [8]. These evaluations can be used for debugging purposes when implementing AD capabilities for additional reaction types or chemical source term models.

For a detailed description of the equations presented in this paper, and for additional information of how to use TChem please see the TChem's GitHub repository [9].

Chemical Reactor Models TChem provides three canonical reactor models: homogeneous gas-phase reactor (0D ignition), continuously stirred tank reactor (CSTR) and plug-flow reactor (PFR). The problems are time-dependent and unsteady. The solution of a stiff time ODE/DAE system requires an adaptive time integration scheme for its accuracy and efficiency. Here again we connect to the TINES library which provides a 2nd order Trapezoidal Backward Difference Formula (TrBDF2) with local error estimation. To solve multiple ODE/DAE problems in parallel, different initial value problems independently proceed with a different, adaptive time step sizes. In this paper, we illustrate a screening study using a PFR model.

3. Results and Discussion

In this section, we discuss the TChem capabilities to evaluate kinetic model properties in parallel and to solve canonical reactor examples. First, we present the parallel performance on CPU and

GPU for multiple evaluations of the specific heat capacity function (lightweight evaluations) and the net production rate function (computationally intensive). Next, we compare the relative time complexity to solve canonical reactor examples using both analytical and numerical Jacobians. Finally, we presented the results for an ensemble of plug reactor model runs done in parallel.

3.1 Numerical Experiments

We first demonstrate the parallel performance of the TChem on a heterogeneous computing architectures i.e., Intel Xeon Gold 62348R 2x28 @2.2GHz accelerated by the NVIDIA GPU V100. A single version of Kokkos implementation is used for both CPU and GPU with Kokkos default (architecture specific) execution policy. We use the GRIMech 3.0 [10] gas kinetic model that include 53 species and 325 elementary reactions. Figure 2 shows the performance of the batch parallelism for the evaluation of thermodynamic and chemical source term properties. On the left, the calculation of heat capacity of the mixture represents a lightweight computational kernel. On the right, the calculation of the net production rate is a computationally intensive process that requires a much larger number of evaluations. The massively parallel nature of the GPU requires a larger number of samples to gain its parallel efficiency. For lightweight computations, the GPU start performing better when computing around 10k samples simultaneously and gains 4x speedup for 1M samples. On the other hand, the species net production rate evaluations, in the right frame, requires just 400 samples for the GPU to perform better than the CPU. When increasing the number of evaluations to 1M samples, the GPU version of the code displays 7x speedup against the CPU. We also compare the performance of the team parallel implementations against the batch parallel implementation that maps a single thread to a single instance of the problems. Although the 1M samples are large enough to make all GPU threads busy, the performance is much slower compared to the team parallel batch implementation. This is largely due to the efficient memory operations of the team parallel implementation, which allows coalesced memory access to its state vector and other work arrays.

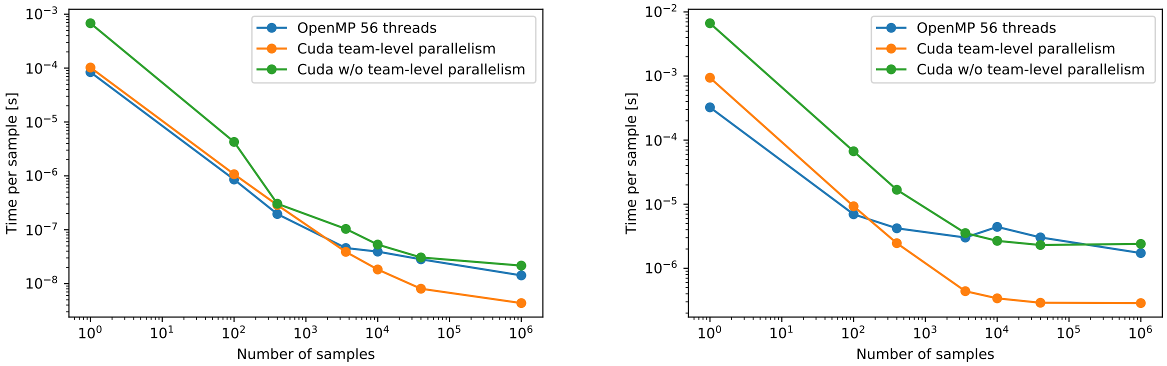

Figure 2: Parallel performance for computations using the GRIMech 3.0 model: (left) specific heat capacity, (right) species net production rate.

Next, we compare the time complexity of the analytical and SACADO-based AD Jacobian evaluations against computations using numerical derivatives. The results shown in Figure 3 are

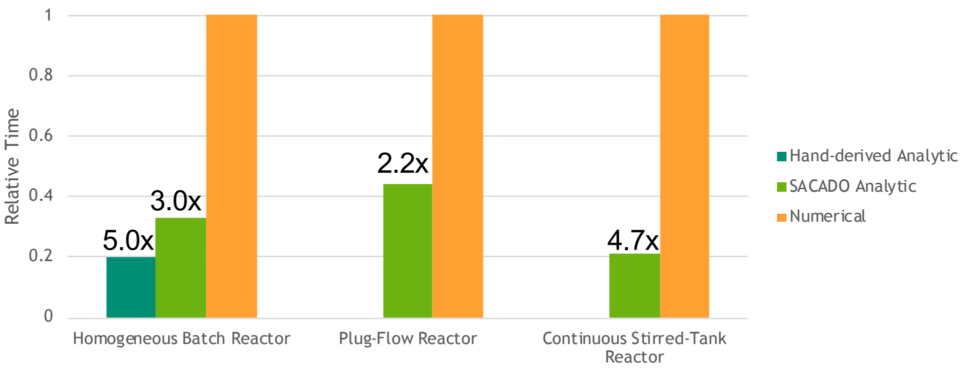

Figure 3: Relative time complexity of analytic Jacobian compared to numerical Jacobian (forward difference scheme).

normalized with the amount of time it takes to compute Jacobian information using finite differences. We use a forward difference scheme to evaluate derivatives numerically. This requires $O(N_{spec})$ times source term evaluations. The AD evaluations via the SACADO library require the only one evaluation of the chemical source terms. The derivatives are updated by chain rules during the evaluation. This reduces the complexity for computing the derivatives as the stoichiometric matrices for most kinetic models of practical interest are typically sparse. The resulting AD-based Jacobian calculation is typically a few times faster than one obtained with finite differences. For the homogeneous batch reactor problem, we have implemented analytical expressions for the requisite derivatives of the chemical source terms. The SACADO-based Jacobian is slower comparing to the version computed via the analytic expressions. However it does not consume an extensive to derive the requisite formula and is robust to inherent implementation errors. Furthermore, as new reaction types become available, the AD-based derivatives are available immediately without extra effort to derive, implement, and test analytical derivatives.

3.2 Chemical Reactor Models:Plug Flow Reactor

Last, we use PFR to demonstrate the batched parallel interface available in TChem. In this example, we design a screening study for four variables: gas velocity at the PFR inlet, equivalence ratio at the same location, the pre-exponential factor for gas-phase reaction No. 84, and the pre-exponential of surface reaction No. 1 using the GRIMech 3.0 [10] gas kinetic model and the surface kinetic model developed by Quinceno [11]. The first two variables are operating condition parameters for a set of runs using a mixture of CH_4 and air. The third variable is a model parameter from the GRIMech 3.0 model that correspond to the pre-exponential factor for the reaction $OH + H_2 \leftrightarrow H + H_2O$. The fourth variable is the pre-exponential factor for the surface reaction $H_2 + 2Pt \rightarrow 2HPt$. We create 81 samples using the minimum, nominal, and maximum values for each parameter provided in Table (1).

TChem can run this problem in parallel using either OpenMP(CPUs) or CUDA(GPUs). For this particular case, we ran the 81 samples on the CPU. We used the same numerical settings for running all samples (see Table (2)), but the samples run independently. The TINES documentation [6] provides more details for the knobs used to control the numerical integration of ODE/DAE systems. The geometry of the PFR configuration is given by the following parameters: length 2.5×10^{-2} m,

Table 1: Range of parameter used in the multi sample simulations of the PFR

Variable	Nominal value	Min value	Max value
Equivalence ratio(ϕ)	1	0.7	1.6
Velocity	0.1	0.05	0.2
Modifier gas reaction No 84	1	1e-4	1e4
Modifier surface reaction No 1	1	0	10

Table 2: Numerical setting for the PFR simulations

Time Step Settings		Time Step Adjustment		Newton Solver		
Min	Max	Relative tolerance	Absolute tolerance	Max iterations	Relative tolerance	Absolute tolerance
1e-20	1e-4	1e-6	1e-12	20	1e-18	1e-8

cross area 5.3×10^{-2} m^2, and the active perimeter is 2.6×10^{-2} m.

Screening studies are an useful precursor for sensitivity or uncertainty quantification analyses. They help define the ranges for the parameters that control the model and identify combinations of parameter values where the model exhibits unexpected behaviors, e.g. lack of convergence or nonphysical results. To verify the simulation output, we are going to plot a simulation results for the nominal sample in Figure (4) and select results for all the samples in Figure 5.

Figure 4 presents the sample corresponding to the nominal values given in Table 1. TINES adjusts the advancing time step size, shown in the top right plot, at spatial locations when simulations variables have a bigger rate-of-change (see top left plot for the temperature and the velocity) specifically at the entry location (0 m) and at 5.2×10^{-3} m. The plot for the site fraction, on the bottom left, shows that surface species have a sharp variation at the inlet and at 5.2×10^{-3} m (ignition point), where the surface becomes almost empty (Pt ≈ 1). The plot for the mass fraction shows that the gas species have sharp changes at the same locations. The changes between these locations are likely due to surface chemistry. Beyond the ignition point, the simulation reaches stable state as most of the fuel is consumed in a narrow range.

Figure 5 shows the profiles of temperature, advancing step, mass fraction of O_2, and site fraction of Pt for all 81 samples. The profile for multi samples is similar to the profiles that we described in detail in Figure (4). The top right plot shows the time step adapts independently across the samples run simultaneously by TChem. Some samples do not reach stable state by 2.5e-2 m.

4. Conclusions

TChem is a software library to compute thermodynamic properties, chemical source terms and Jacobian information for gas-phase and surface chemistry models. It also provides chemical reactor models for applications that involve chemical reactive flows on heterogeneous computing platforms. The code is designed to solve the many problems in a batch parallel fashion and ported to multiple computing platforms using Kokkos. To our best knowledge, TChem is the first open source software that can solve the chemical kinetic models on mixed CPU/GPU platforms. We present its parallel performance for both lightweight computations, e.g. thermodynamic properties, and computationally intensive evaluations, e.g. species production rates. For GPU computa-

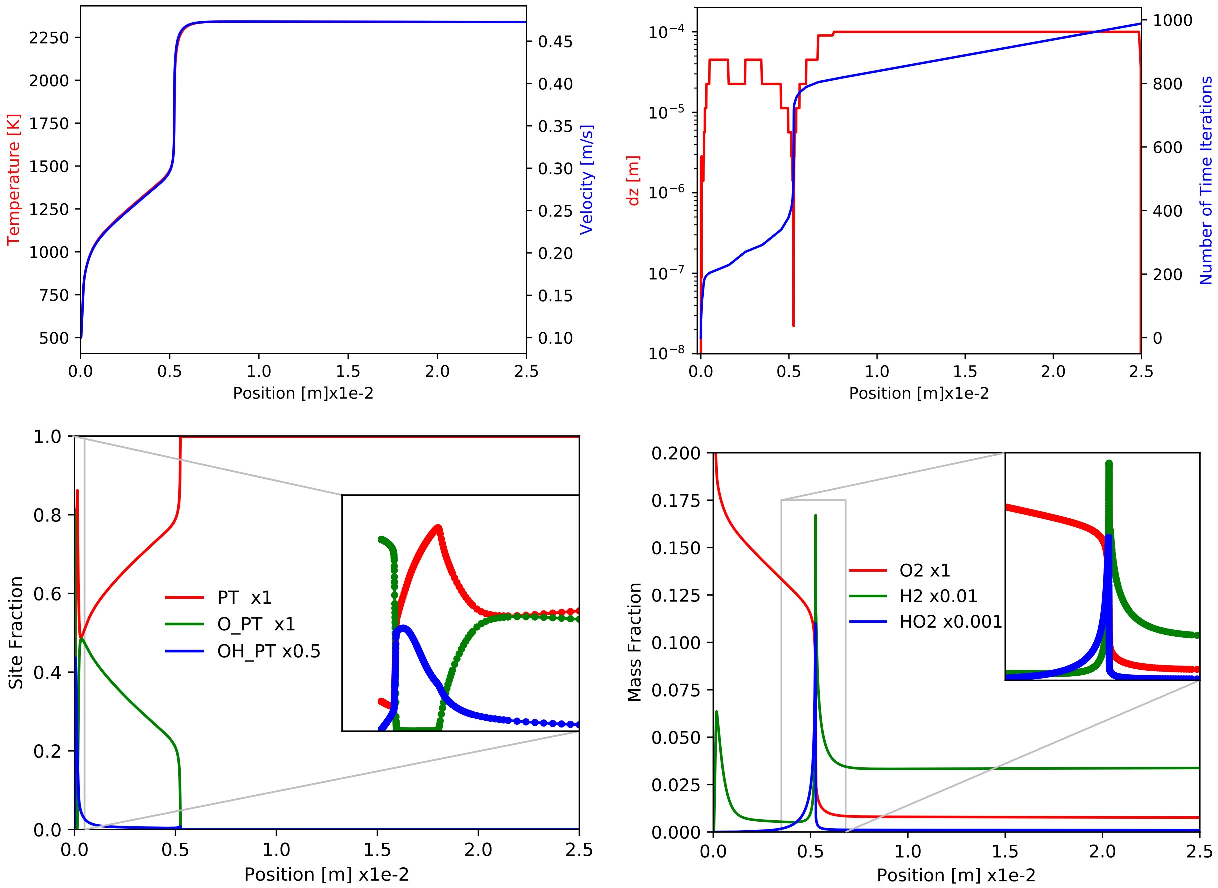

Figure 4: PFR profiles for one sample using nominal values from Table (1). Top left plot: Temperature(red) and velocity(blue) profiles. Top right: advancing step or "time step" (red) and number of time iterations(blue) profiles. Bottom left: Site fractions for Pt (free fraction on the surface), OPt, and OHPt profiles. Bottom right: Mass fraction for O2, H2, and HO_2 profiles.

tions the library tends to perform better for a large number of samples. For 10k samples, the GPU gains 4x speedup for the source term evaluations. We evaluate the performance the analytic Jacobian evaluations via the automatic differentiation tools provided by the SACADO library. This approach provide evaluations that are robust to numerical errors and is less expensive than numerical Jacobians evaluations as the derivatives are updated following the sparse reaction model. We demonstrate the batched parallel interface provided by TChem on an example study using a canonical plug flow reactor model.

Acknowledgements

This work is supported as part of the Computational Chemical Sciences Program funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, Chemical Sciences, Geosciences and Biosciences Division. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National

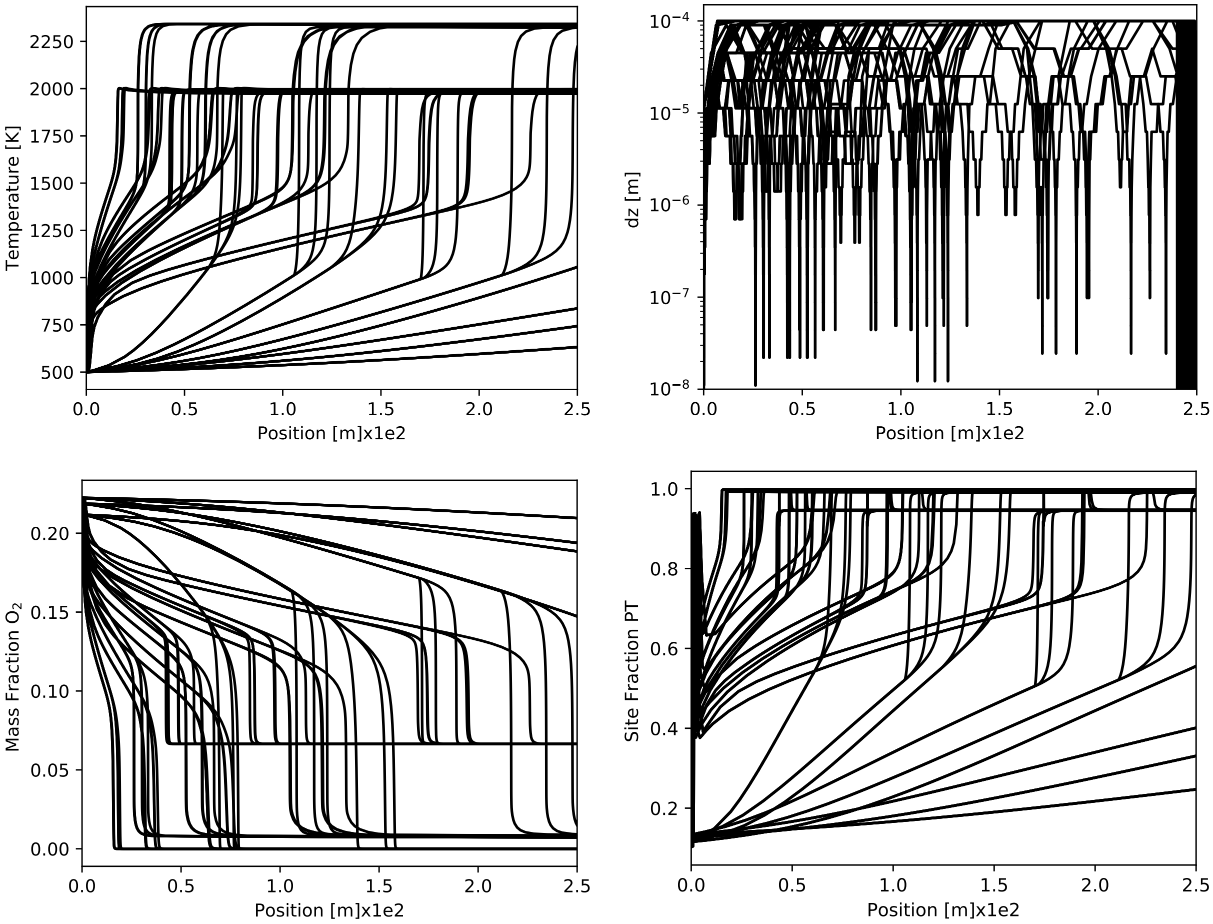

Figure 5: PFR profiles for multi samples varying inlet gas equivalent ratio (ϕ), inlet gas velocity, pre-exponential parameter for gas reaction No 84 ($OH + H_2 \leftrightarrow H + H_2O$), and pre-exponential parameter surface reaction No 1 ($H_2 + 2Pt \rightarrow 2HPt$). Top left: Temperature profiles. Top right: Advancing step or "time step" profiles. Bottom left:O2 Mass fraction profiles. Bottom right: Pt site fraction profiles

Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

[1] D. Goodwin, H. K. Moffat, and R. L. Speth, Cantera: An Object-oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes. Version 2.2.0, (2015).

[2] *Ansys Chemkin-Pro*, `https://www.ansys.com/products/fluids/ansys-chemkin-pro`, Accessed: 2021-06-02, 2021.

[3] K. Kim, O. Diaz-Ibarra, C. Safta, and H. N. Najm, TChem v2.0 - A Software Toolkit for the Analysis of Complex Kinetic Models, tech. rep. Report No. SAND2020-10762, Livermore, CA: Sandia National Laboratories, 2020.

[4] H. C. Edwards, C. R. Trott, and D. Sunderland, Kokkos: Enabling manycore performance portability through polymorphic memory access patterns, Journal of Parallel and Distributed Computing 74 (2014) 3202–3216.

[5] B. J. McBride, S. Gordon, and M. A. Reno, Coefficients for Calculating Thermodynamic and Transport Properties of Individual Species, tech. rep. Report No. NASA TM-4513, NASA, 1993.

[6] *TINES*, `https://github.com/sandialabs/Tines`, Accessed: 2021-04-02, 2021.

[7] E. T. Phipps, R. A. Bartlett, D. M. Gay, and R. J. Hoekstra, Large-Scale Transient Sensitivity Analysis of a Radiation-Damaged Bipolar Junction Transistor via Automatic Differentiation, Advances in Automatic Differentiation Springer Berlin Heidelberg, Berlin, Heidelberg, (2008), pp. 351–362.

[8] D. E. Salane, Adaptive routines for forming Jacobians numerically, SAND86 (1986).

[9] *TChem*, `https://github.com/sandialabs/TChem`, Accessed: 2021-04-02, 2021.

[10] G. P. Smith, D. M. Golden, M. Frenklach, N. W. Moriarty, B. Eiteneer, M. Goldenberg, C. T. Bowman, R. K. Hanson, S. Song, W. C. G. Jr., V. V. Lissianski, and Z. Qin, *GRI-Mech v3.0*, `http://www.me.berkeley.edu/gri_mech`, Accessed: 2020-08-26, 2011.

[11] R. Quiceno, J. Pérez-Ramírez, J. Warnatz, and O. Deutschmann, Modeling the high-temperature catalytic partial oxidation of methane over platinum gauze: Detailed gas-phase and surface chemistries coupled with 3D flow field simulations, Applied Catalysis A: General 303 (2006) 166–176.